AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. A method for optimizing an executable comprising:

determining a set objects, included in a plurality of objects, eligible for migration by identifying

objects of the plurality of objects that are not accessible from multiple processors, and

objects in an equivalence set of aliased objects that does not further include an object that may be accessed by multiple processors;

migrating the set of objects eligible for migration from a first memory to a second memory, wherein migrating includes

changing residence of the set of objects determined to be eligible for migration, and

changing accesses of the set of objects having their residence changed;

determining alignment of the migrated set of objects; and

eliminating redundant initialization code of the set of objects.

2. The method for optimizing an executable of claim 1, wherein the plurality of objects are variables.

3. (Cancelled)

4. The method for optimizing an executable of claim 1, the determining alignment further comprising:

analyzing the migrated objects by forward disjunctive dataflow analysis;

determining a minimum alignment necessary for each migrated object; and

setting the minimum alignment necessary for each migrated object.

5. The method for optimizing an executable of claim 1, wherein the first memory is an external memory and the second memory comprises a plurality of indexed registers residing in a microengine.

6. (Currently Amended) A processing device comprising:

an optimizer to determine a set of ~~variables~~ **objects**, included in a plurality of ~~variables~~ **objects**, eligible for migration by identifying ~~variables~~ **objects** of the plurality of ~~variables~~ **objects** that are not accessible from multiple network processors, and ~~variables~~ **objects** in an equivalence set of aliased ~~variables~~ **objects** that does not further include a**n** ~~variables~~ **object** that may be accessed by multiple network processors, change location of the set of ~~variables~~ **objects** that are determined to be eligible for migration, and migrate the set of objects from an external memory of a network processing device to a plurality of registers coupled to a processor, the optimizer further to align and eliminate redundant initialization code of the set of objects.

7. The processing device of claim 5, wherein the plurality of registers are indexed.

8. The processing device of claim 5, wherein the plurality of objects are variables.

9. The processing device of claim 5, wherein the migrated plurality of objects are not shared by the processor and at least one other processor.

10. The processing device of claim 5, wherein the network processing device is a router.

11. (Currently Amended) An optimizer system for network processors comprising:

a processor,

a first memory coupled to the processor;

~~a display coupled to the processor;~~

a compiler to migrate a set of objects, included in a plurality of objects, from a second memory to a plurality of indexed registers in a network processor, the compiler further to align and eliminate redundant initialization code of the migrated set of objects;

a determiner to determine the set of objects of the plurality of objects eligible for migration by identifying

objects of the plurality of objects that are not accessible multiple processors in a network device, and

objects in an equivalence set of aliased objects that does not further include an object that may be accessed by multiple processors in a network device

a migrator to change residence of the set of objects determined to be eligible for migration; and

an accessor to change accesses of the set of objects having their residence changed.

12. (Cancelled)

13. The optimizer system for network processors of claim 11, wherein the second memory is external to the plurality of processors.

14. The optimizer system for network processors of claim 11, wherein the plurality of objects are variables.

15. The optimizer system for network processors of claim 11, wherein the second memory is external to the plurality of indexed registers.

16. A machine-readable storage medium containing instructions that, when executed, cause a machine to:

Determine a set of variables, included in a plurality of variables, eligible for migration by identifying

variables of the plurality of variables that are not accessible from multiple network processors, and

variables in an equivalence set of aliased objects that does not further include a variable that may be accessed by multiple network processors;

change location of the set of variables that are determined to be eligible for migration;

migrate the set of variables from a first memory to a plurality of indexed registers;

align the migrated set of variables; and

eliminate redundant initializations to a base address register.

17. (Cancelled)

18. The machine-readable storage medium of claim 16, further comprising instructions that, when executed, cause a machine to:

analyze the migrated variables by forward disjunctive dataflow analysis; and

determine a minimum alignment necessary for each migrated variable.

19. The machine-readable storage medium of claim 16, further comprising instructions that, when executed, cause a machine to:

set the minimum alignment necessary for each migrated variable.

20. The machine-readable storage medium of claim 16, further comprising instructions that, when executed, cause a machine to:

compile source code to migrate the plurality of variables from the first memory to the plurality of indexed registers.